**112學年 第2學期 編譯器設計 小專題報告(1) 日期:2024/4/14**

班級: 資工三 學號: B1043003 姓名: 陳麒安

1. **詞彙分析器(Scanner) (共1題，100分，滿分100分)**

請於2024/4/14晚上12:00前透過數位學習M園區繳交

請以指定教材3.5小節內容做為參考進行以下的實作，每小題須**貼關鍵程式碼或腳本，截圖呈現結果並文字說明。(注意: 請先使用sudo hostname [學號]指令修改主機名稱為學號再截圖，否則不予計分。)**

1. 請自選一段至少10行以上的任何語言程式碼作為詞彙分析器之輸入(source program)。 (25分)

print('Hello world')

num1 = 20

num2 = 3

add = num1 + num2

sub = num1 - num2

mul = num1\*num2

div = num1/num2

print(add)

print(sub)

print(mul)

print(div)

一張含有 軟體, 文字, 多媒體軟體, 電腦圖示 的圖片

自動產生的描述

圖一、python程式碼及執行畫面

1. 以程式碼語言的特性撰寫FLEX程式以定義詞彙單位(lex.l)。 (25分)

選擇Ubuntu系統進行實作。

Lex 檔案內容：  
%{

#include <stdio.h>

int keyword\_count = 0;

int identifier\_count = 0;

int integer\_count = 0;

int string\_count = 0;

int operator\_count = 0;

int unrecognized\_count = 0;

%}

%%

print { keyword\_count++; printf("Keyword: print\n"); }

[0-9]+ { integer\_count++; printf("Integer: %s\n", yytext); }

[a-zA-Z\_][a-zA-Z0-9\_]\* { identifier\_count++; printf("Identifier: %s\n", yytext); }

\".\*\" { string\_count++; printf("String: %s\n", yytext); }

[+\-\*/=] { operator\_count++; printf("Operator: %s\n", yytext); }

[ \t\n] ; // Ignore whitespace and newline

. { unrecognized\_count++; printf("Unrecognized: %s\n", yytext); }

%%

int main() {

yyin = fopen("test.py", "r"); // Open input file

if (!yyin) {

fprintf(stderr, "Failed to open input file.\n");

return 1;

}

yylex(); // Perform lexical analysis

fclose(yyin); // Close input file

printf("\*\*\*\*\*\*\*\*\*\* Output Count \*\*\*\*\*\*\*\*\*\*\n");

printf("Keyword count: %d\n", keyword\_count);

printf("Identifier count: %d\n", identifier\_count);

printf("Integer count: %d\n", integer\_count);

printf("String count: %d\n", string\_count);

printf("Operator count: %d\n", operator\_count);

printf("Unrecognized count: %d\n", unrecognized\_count);

return 0;

}

1. 試用FLEX編譯器產生lex.yy.c程式原始碼，並以你的理解來分析說明。(25分)

首先，在終端機中執行 “flex lex.l” 命令來編譯Flex程式並產生C語言原始檔。接著使用 “gcc lex.yy.c -o lexer -lfl” 命令將 lex.yy.c 編譯成可執行檔。最後，執行 “flex lex.l” 命令將Python程式碼檔案作為輸入，得到運行 lexer的結果。

一張含有 軟體, 多媒體軟體, 電腦圖示, 繪圖軟體 的圖片

自動產生的描述

圖二、撰寫FLEX程式並產生執行檔

當使用“gcc lex.yy.c -o lexer -lfl”處理 lex`檔案時，它會生成一個C程式檔案，其中`yy`是由`lex`工具產生的前綴，而`.c`則表示這是一個C程式檔案。

在`lex.yy.c`中，看到一些以`yy`為前綴的函式和變數，這些是由`lex`工具自動生成的。其中一些函式是`yylex()`，它是lexical analyzer的入口函式，用於從輸入文本中讀取字符並根據我在`lex`檔案中定義的規則進行匹配和處理。另外一些函式則是用來輔助`yylex()`函式的，比如用來返回匹配到的token的內容的`yytext`，以及用來向`yylex()`提供輸入字符的`yyinput()`。

在`lex.yy.c`中，還會看到由`%{ ... %}`區塊中的程式碼直接複製到頂部，以及由`%%`區塊中的程式碼轉換為`yylex()`函式的規則和動作。

以我的觀察來看，`lex.yy.c`是一個可以用來對輸入文本進行lexical analysis的C程式。它包含了根據我在`lex`檔案中定義的規則自動生成的程式碼，這些規則用於識別和處理輸入文本中的不同token，並且可以根據這些token進行後續的處理。

1. 試以lex.yy.cc編譯而成的程式作為詞彙分析器，，並以自選程式碼片段作為輸入，產出詞彙單位序列與詞彙統計。(25分)

一張含有 文字, 螢幕擷取畫面, 多媒體軟體, 軟體 的圖片

自動產生的描述

圖三、執行 ./lexer < test.py 之結果

執行結果：  
b1043003@b1043003:~$ ./lexer < test.py

Keyword: print

Unrecognized: (

Unrecognized: '

Identifier: Hello

Identifier: world

Unrecognized: '

Unrecognized: )

Identifier: num1

Operator: =

Integer: 20

Identifier: num2

Operator: =

Integer: 3

Identifier: add

Operator: =

Identifier: num1

Operator: +

Identifier: num2

Identifier: sub

Operator: =

Identifier: num1

Operator: -

Identifier: num2

Identifier: mul

Operator: =

Identifier: num1

Operator: \*

Identifier: num2

Identifier: div

Operator: =

Identifier: num1

Operator: /

Identifier: num2

Keyword: print

Unrecognized: (

Identifier: add

Unrecognized: )

Keyword: print

Unrecognized: (

Identifier: sub

Unrecognized: )

Keyword: print

Unrecognized: (

Identifier: mul

Unrecognized: )

Keyword: print

Unrecognized: (

Identifier: div

Unrecognized: )

\*\*\*\*\*\*\*\*\*\* Output Count \*\*\*\*\*\*\*\*\*\*

Keyword count: 5

Identifier count: 20

Integer count: 2

String count: 0

Operator count: 10

Unrecognized count: 12